2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)
Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. Aug 18-19, 2023

Play Store App Analysis & Rating Prediction using Classical ML Models & Artificial Neural Network

Bhimasen Moharana
Department of Computer Science and Engineering
Lovely Professional University
Punjab, India
bhimasen.moharana@gmail.com

Bhramara Bar Biswal
P.G. Department of Computer Science & Applications
College of Engineering
Bhubaneswar, India
b.b.biswal1969@gmail.com

Snehasis Dey
Electronics & Telecommunication Engineering
College of Engineering
Bhubaneswar, India
snehasis9dey@gmail.com

Manas Kumar Rath
Department of Computer Application
Maharaja Sriram Chandra Bhanja Deo University
Baripada, India
manasrath@gmail.com

Shobhan Banerjee
Department of Engineering Technology
Birla Institute of Technology and Science
Pilani, India
shobhanbanerjee3@gmail.com

***Abstract*—Each app available in the App Store or Play Store has various aspects linked to it such as its version, category, number of installations, genre, etc. which makes it unique, robust & popular. The reviews and ratings given by end users play an important role for both the developers and other users for the performance and survival of the app in the marketplace. Reviews are just a description for justification of the rating given by a user & the rating plays an important role in the first look. In this paper, we have used the Play Store Analysis Dataset to descriptively analyze the various attributes present in it and create models that predict the rating of an app, given its specifications. The models have been compared based on a common accuracy metric based on which their reliability can be judged and used in real-time rating predictions for a new application in the store for which a certain amount of data is available over some period.**

Keywords—Play Store App Analysis, Rating Prediction, ML Classification Models, Descriptive Statistics, Model Comparison

I. INTRODUCTION

The apps available in Apple App Store or the Google Play Store are evaluated based on their reviews and ratings on the store. The reviews as mentioned above are just a mere descriptive justification of the rating given by an individual. The review analytics forms an important application of text and Natural Language Processing that is beyond the scope of this paper for now. But the first impression of the app is formed by the overall rating out of five. Hence rating prediction is an important application in the continuous numeric domain.

Authors in [1] have worked on the Google Play Store dataset, in which they concluded that the data available is not sufficient to train an ML model & hence more data needs to be gathered in the future. In [2], the authors performed sentiment analysis on popular applications in Google Play Store using RBF kernel in SVM & acquired 73.97% accuracy. The authors in [3] have proposed SERS – a ranking scheme based on the security aspects of an app. We see an improved cluster analysis algorithm in [4] for datasets of multi-dimensionality where the data has been segregated into subsets, followed by which cluster analysis has been done on them individually and then integrated on the whole in the same data space. The authors have worked towards finding a reliable health app in [5] in Arabic in Health & Fitness as well as medical categories from the Android App Store.

App Store content organization has been investigated in [6], where the developers build and upload the app & the users browse the store and download them, where the apps have been organized by top apps chart & new apps chart IOS App Store. A numeric rating approach based on sentiment analysis and an optimized probabilistic approach has been proposed by the authors in [7], in a diverse corpus where the targets are of different categories. Authors in [8] have claimed to predict the success of an android application based on user ratings and the number of installations before even it gets uploaded to the App Store. Authors in [9] have performed scraping of data & content visualization for analytics of the Google Play Store data by studying the relationship between attributes & visualizing them in CIRCOS. In [10], the indicators have been studied that can affect reviews, ratings & the number of downloads for an app, using Exploratory Analysis of Data and their visualization methods.

In this paper, we've used the Google Play Store Apps dataset [11], to perform our research on the available data. We first performed an exploratory data analysis on this dataset where we tried to answer certain questions & address certain issues, followed by which we cleaned the data. Then after, we posed the problem of predicting the rating as a rating classification task by separating the real numbered values under 5 bins.

II. DATA DESCRIPTION

The dataset consists of 10841 rows and 13 columns. Looking into the data distribution, we see that most of the data in each of the multiple data points corresponding to the same app is the same, apart from reviews. Presumably, this might be an update error, where multiple incoming reviews might've prompted multiple entries. To remedy this, we picked one datapoint for each App with the highest number of reviews (logically, the most recently entered, thus the most accurate datapoint for the App).

2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA) | 979-8-3503-0426-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCUBEA58933.2023.10391960

III. Exploratory Data Analysis

We removed the duplicate entries from the *App* Column and found out that there are 9660 unique apps present in that column. 34 unique *categories* were found of which one of them was even named '1.9'. The top five categories along with their value counts have been shown in the figure below.

FAMILY	1873
GAME	958
TOOLS	829
BUSINESS	419
MEDICAL	396

Fig. 1. Top 5 categories & their counts

The *ratings* range from 0 to 5, with an average rating of 4.17 for an app. The rating distribution can be visualized in the figure below.

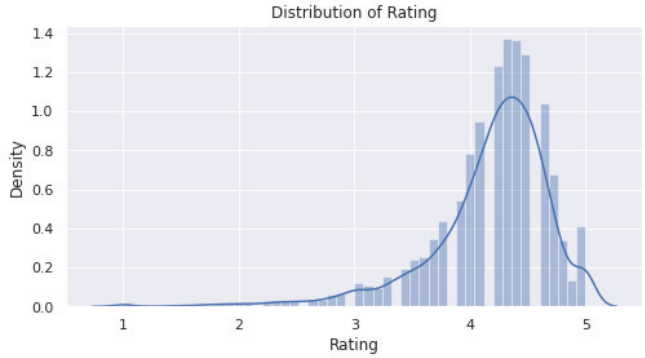

Fig. 2. Rating distribution

Looking into the *installations*, we saw that the average number of installations is around 7.8M for a given application with a minimum value of even 0 for a few of them. We found 21.6M *reviews* on average present for the apps. The *size* of apps varied from several kilobytes to several megabytes. Also, there were certain apps whose sizes varied from device to device. A significant number of entries had size as null. We can't simply delete them, so they have been imputed with the mean value, where the mean size of an app is 20 MB.

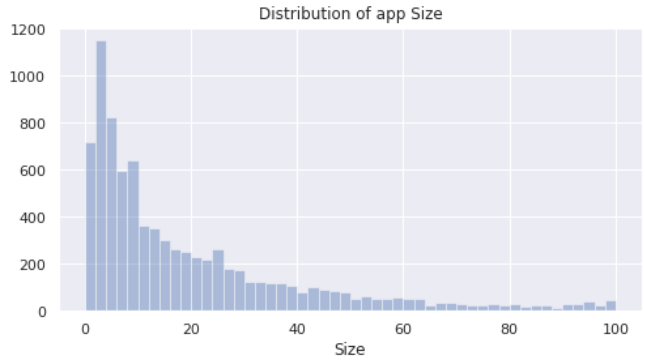

Fig. 3. App Size distribution

Figure 3 shows the distribution for app size. The apps are of two types – free and paid. There are a total of 8902 free apps & 756 paid apps. However, in the previous versions [1] we saw that 95.7% of the apps were free and 4.3% of the apps were paid. This was converted into a categorical attribute for eventual computation. The average *price* distribution for the current dataset in percentages has been shown in the pie chart below.

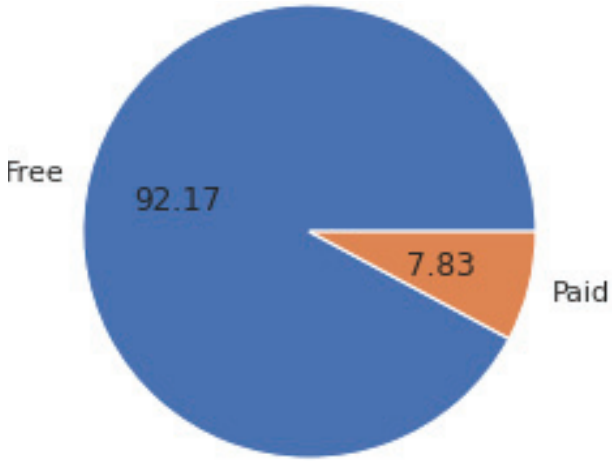

Fig. 4. Type distribution

The average price of paid apps in the Play Store is $1.097. *Content rating* is another attribute that specifies the age range from which the rating is done. The age ranges and their counts are shown in the figure below.

Everyone	7903
Teen	1036
Mature 17+	393
Everyone 10+	322
Adults only 18+	3
Unrated	2

Fig. 5. Content Ratings & the counts

Since there are only two unrated data points, we can remove these from our dataset. This was also converted to a categorical attribute. The next attribute is *genres* – which were 48 in total throughout our dataset. The top 10 genres are shown in figure 6.

Tools	828
Entertainment	591
Education	580
Business	419
Medical	396
Personalization	376
Productivity	374
Lifestyle	370
Finance	345
Sports	335

Fig. 6. Top 10 genres

Android version and the *last update* were two more attributes whose values are justified by the attribute names themselves. Some of the NAN versions were found in the Android version and those were imputed using the mode values of these versions.

IV. Data Preprocessing

Some of the missing data values were imputed during the EDA process as mentioned in the previous section. But while skimming through the data, we found that still there were a lot of missing values present in between. Also, there were some erroneous data like the '1.9' category as mentioned earlier. There was only a single row present corresponding to this category hence we removed that single category from the dataset. In addition to that, there were 1229 points where the size of the apps varied with the device. Hence, we first converted them into null values & replaced these values with the mean of the size of the App's Genre. We chose Genre over category as it is more atomic in nature. For a single data point, we had a null value for the type, but since the price corresponding to that was zero, hence we imputed it with free. Also, there were 1474 null ratings whose values can be predicted after model creation and validation, but for training and testing, we simply dropped those rows.

V. Correlation Analysis

Considering the relevant attributes from the dataset, we have now performed a correlation analysis between them to see their importance. The correlation matrix is shown in Figure 7.

Since installs and reviews are in orders of millions, hence we considered their common log values.

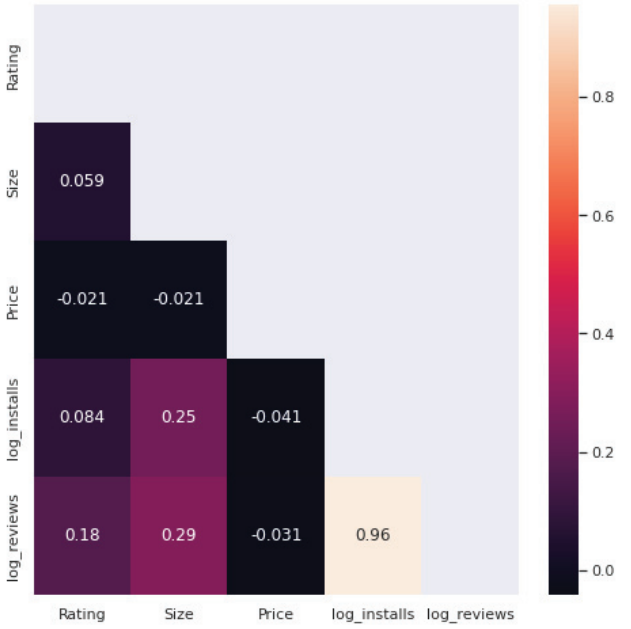

Fig. 7. Correlation Matrix

The following observations can be made from the heatmap above-

- Installations and reviews are highly correlated indicating that the app with more reviews will generally have more installations.
- Size is weakly correlated to installations & rating is weakly correlated to reviews.

Category and genres were also found to be highly correlated with a value of 81%, which means that we should choose one of the columns to keep the meaningfulness of the data maximal.

VI. Statistical Inferences

Diving deep into the data, we can answer certain questions & derive certain conclusions about the data. They have been summarized in the below-mentioned points.

- Games as a category are the most popular, having the largest number of installations.
- Hungry Shark Evolution is the most popular game where the size is greater than 100 MB.
- Assuming that the app with the highest number of reviews will have the highest number of downloads, the top five downloaded apps are as follows:
  - Facebook
  - WhatsApp Messenger
  - Instagram
  - Messenger – Text & Video Chat for free
  - Subway Surfer
- Hence, Facebook also has the largest number of reviews, i.e., 78M.

The features were separated based on their nature into categorical and numeric features. We used the χ^2-test to identify the best categorical features. The feature scores for the categorical features have been shown in figure 8 below.

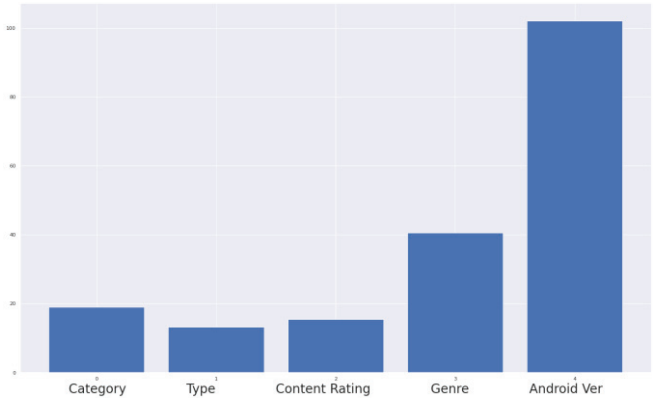

Fig. 8. Feature importance for categorical features

Analysis of Variance (ANOVA) was performed to identify the best numeric features. The feature scores for the numerical features have been shown in figure 9 below.

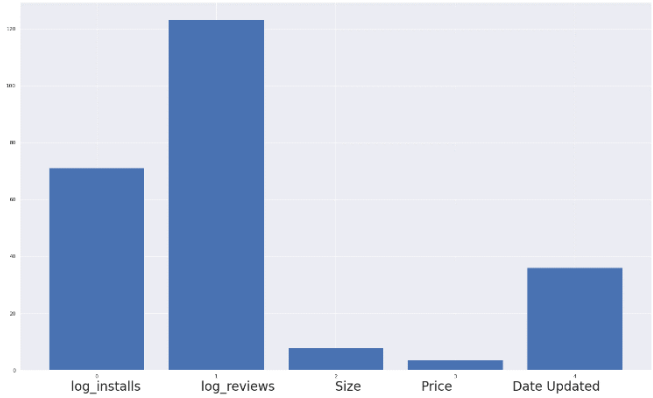

Fig. 9. Feature importance for numerical features

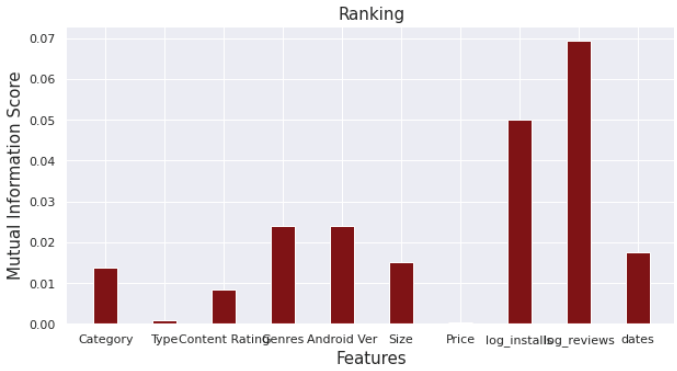

Fig. 10. Mutual Information Scores

Figure 10 shows the mutual information scores for all the attributes, based on which the most informative features were found to be – log_reviews > log_installs > dates > genres > category > android version > size > content rating > price > type.

VII. Machine Learning Classification Models

We considered the top six attributes to train various classical ML models. Let's see if we are getting the desired level of accuracy or not through these models or else, we shall have to look at some NNs for classification.

We used Dataiku to perform a comparative analysis between various Machine Learning models, the ROC-AUC values for which have been shown in Figure 11. The data were randomly split in a 4:1 train-test ratio, i.e., 6556 samples were used for the training set and 1639 samples for testing the model.

Fig. 11. ROC-AUCs for various ML models

From figure 11 we can infer that the Logistic Regression model was the best based on the area under the receiver operating characteristic curve. But the accuracies of these models were not quite satisfactory, which have been summarized below.

Algorithm	Accuracy
Random Forest	71.6081%
Logistic Regression	71.6081%
Decision Tree	71.6081%
K Nearest Neighbours	71.6451%
Support vector Machine	71.6081%

Fig. 12. Accuracy Classifications of ML Models

We increased the n_estimators value for KNN to 60, where we saw slightly better performance than the other models. In the next section, we tried some Deep Neural Networks to see whether the performance can be enhanced or not.

VIII. Deep Learning Classification Model

We trained an artificial neural network to see if the performance gets enhanced. The visualization of the proposed architecture of the ANN is shown in figure 13. It consists of three hidden layers of 12, 10 & 8 neurons each. The input layer matches that of the input dimensionality of 6 neurons and the output layer with 5 neurons. All the layers are densely connected with Rectified Linear Unit activation function in the hidden layers and SoftMax at the output layer.

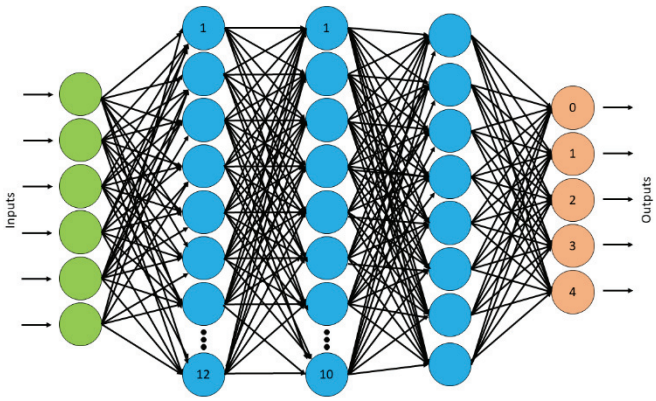

Fig. 13. The ANN Architecture

Before feeding the data to the network, we performed min-max scaling to avoid any sort of data leakage. The data was split in a 2:1 ratio, hence giving us 5490 train samples & 2705 test samples. Figure 14 shows the model summary of the ANN architecture.

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_19 (Dense)            (None, 12)                84

 dense_20 (Dense)            (None, 10)                130

 dense_21 (Dense)            (None, 8)                 88

 dense_22 (Dense)            (None, 5)                 45

=================================================================
Total params: 347
Trainable params: 347
Non-trainable params: 0
_________________________________________________________________
```

Fig. 14. ANN Model Summary

The model was compiled intending to minimize the sparse categorical entropy loss using RMSprop optimizer & enhance the accuracy metric. The model was trained up to 15 epochs in batches of 8. The training accuracy & loss are shown in Figure 15.

This model after validation gave us an accuracy of 81.61% on the test set with a loss of 0.157. The heatmap shown in Figure 16 represents the confusion matrix as our classification accuracy metric. We see that there are 498 misclassified points across all the class ratings. But the major ambiguity is created between 4 and 5-star ratings for 425 data points which are almost treated as positive ratings.

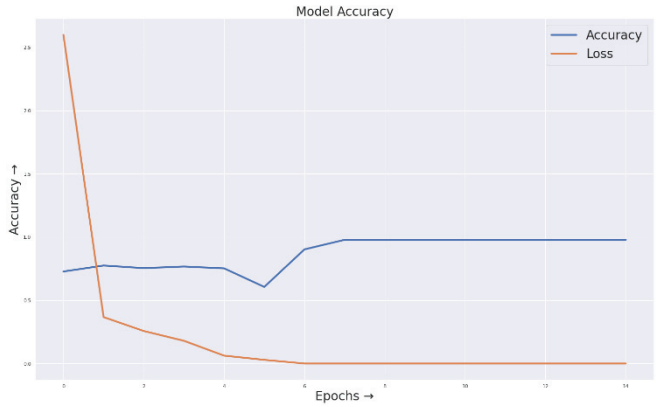

Fig. 15. Training Accuracy & Loss

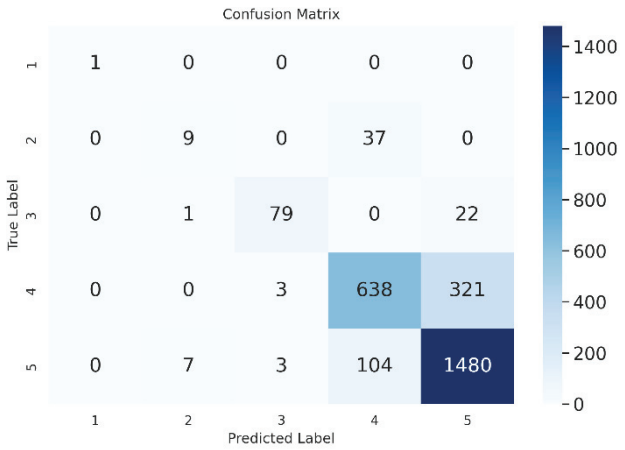

Fig. 16. Confusion Matrix

IX. Results & Conclusion

The accuracies obtained from the classical ML models were not satisfactory, hence we used an ANN model which gave us better accuracy of 81.61%. The major tradeoff in accuracy lay between 4-star and 5-star ratings. But the positive point is that both are considered to be positive ratings from the perspective of an end user. But a few of the points need to be seriously looked at, such as where 7 apps are actually 5-star rated apps, but the architecture predicted them to be 2-star apps. Similarly, 37 apps were actually 2-star apps but were predicted to be 4-star apps.

We can't comment upon the data sufficiency but can certainly try other ways to improve the accuracy of our model by using some LSTM architecture. In addition to that, instead of categorizing our original real-valued target attributes, we can pose this as a regression problem where the exact real-valued rating can be predicted. These things will be addressed in our future works, where we'll certainly enhance the accuracy metric and hence, the quality of our model.

References

[1] P. B. Prakash Reddy and R. Nallabolu, "Machine learning based Descriptive Statistical Analysis on Google Play Store Mobile Applications," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 647-655, doi: 10.1109/ICIRCA48905.2020.9183271.

[2] A. Setiawan and V. C. Mawardi, "Android Application For Analysis Review On Google Playstore Using Support Vector Machine Method," 2022 5th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2022, pp. 331-336, doi: 10.1109/ICOIACT55506.2022.9972122.

[3] N. S. Chowdhury and R. R. Raje, "SERS: A Security-Related and Evidence-Based Ranking Scheme for Mobile Apps," 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 2019, pp. 130-139, doi: 10.1109/TPS-ISA48467.2019.00024.

[4] Y. Shi and D. Brown, "An Attempt to Discover Analytical Information for Multi-Dimensional Data Sets," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2018, pp. 1-5, doi: 10.1109/ICIRCA.2018.8597350.

[5] F. Akbar and L. Fernandez-Luque, "What's in the Store? A Review of Arabic Medical and Health Apps in the App Store," 2016 IEEE International Conference on Healthcare Informatics (ICHI), Chicago, IL, USA, 2016, pp. 413-413, doi: 10.1109/ICHI.2016.77.

[6] S. L. Lim and P. J. Bentley, "Investigating app store ranking algorithms using a simulation of mobile app ecosystems," 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 2013, pp. 2672-2679, doi: 10.1109/CEC.2013.6557892.

[7] M. R. Islam, "Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews," 2014 International Conference on Electrical Engineering and Information & Communication Technology, Dhaka, Bangladesh, 2014, pp. 1-4, doi: 10.1109/ICEEICT.2014.6919058.

[8] G. M. Muradul Bashir, M. S. Hossen, D. Karmoker and M. J. Kamal, "Android Apps Success Prediction Before Uploading on Google Play Store," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9068071.

[9] R. M. Amir Latif, M. Talha Abdullah, S. U. Aslam Shah, M. Farhan, F. Ijaz and A. Karim, "Data Scraping from Google Play Store and Visualization of its Content for Analytics," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-8, doi: 10.1109/ICOMET.2019.8673523.

[10] Qolbi, Shafira & Zahra, Aishaa & Larasati Anisa Rahma, Intan. (2022). ANALISIS DATASET GOOGLE PLAYSTORE MENGGUNAKAN METODE EXPLORATORY DATA ANALYSIS Analysis of Google Playstore Datasets Using Exploratory Data Analysis Methods.

[11] Prakash, G., & Koshy, J. (2021). Google Play Store Apps [Dataset]. In *Google Play Store App data of 2.3 Million+ applications.* (Version 7). Kaggle. https://www.kaggle.com/datasets/gauthamp10/google-playstore-apps.

[12] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.